General Self-Motivation and Strategy Identification: Case Studies based on Sokoban and Pac-Man

Tom Anthony, Daniel Polani, Chrystopher L. Nehaniv

***Abstract*—We use *empowerment*, a recently introduced biologically inspired measure, to allow an AI player to assign utility values to potential future states within a previously un-encountered game without requiring explicit specification of goal states. We further introduce *strategic affinity*, a method of grouping action sequences together to form 'strategies', by examining the overlap in the sets of potential future states following each such action sequence. Secondly, we demonstrate an information-theoretic method of predicting future utility. Combining these methods, we extend empowerment to *soft-horizon empowerment* which enables the player to select a repertoire of action sequences that aim to maintain anticipated utility.**

We show how this method provides a *proto-heuristic* for non-terminal states prior to specifying concrete game goals, and propose it as a principled candidate model for "intuitive" strategy selection, in line with other recent work on "self-motivated agent behaviour". We demonstrate that the technique, despite being generically defined independently of scenario, performs quite well in relatively disparate scenarios, such as a Sokoban-inspired box-pushing scenario and in a Pac-Man-inspired predator game, suggesting novel and principle-based candidate routes towards more general game-playing algorithms.

***Index Terms*—Artificial intelligence (AI), information theory, Games**

I. Introduction

A. Motivation

"Act always so as to increase the number of choices."
- Heinz von Foerster

In many games, including some still largely inaccessible to computer techniques, there exists for many states of that game a subset of actions that can be considered "preferable" by default. Sometimes it is easy to identify these actions, but for many more complex games it can be extremely difficult. While in games such as Chess algorithmic descriptions of the quality of a situation have led to powerful computer strategies, the task of capturing the intuitive concept of the *beauty* of a position, often believed to guide human master players, remains elusive (1). One is unable to provide precise rules for a beauty heuristic, which would need to tally with the ability of master Chess players to appreciate the structural aspects of a game position, and from this identify important states and moves.

Whilst there exist exceedingly successful algorithmic solutions for some games, much of the success derives from a combination of computing power with human explicitly designed heuristics. In this unsatisfactory situation, the core challenge for AI remains: can we produce algorithms able to identify relevant structural patterns in a more general way which would apply to a broader collection of games and puzzles? Can we create an AI player motivated to identify these structures itself?

Game tree search algorithms were proposed to identify good actions or moves for a given state (2; 3; 4). However, it has since been felt that tree search algorithms, with all their practical successes, make a limited contribution in moving us towards 'intelligence' that could be interpreted as plausible from the point of view of human-like cognition; by using brute-force computation the algorithms sidestep the necessity of identifying how 'beauty' and related structural cues would be detected (or constructed) by an artificial agent. John McCarthy predicted this shortcoming could be overcome by brute-force for Chess but not yet Go [1] and criticising that with Chess the solutions were simply 'substituting large amounts of computation for understanding' (5). Recently, games such as Arimaa were created to challenge these shortcomings (6) and provoke research to find alternative methods. Arimaa is a game played with Chess pieces, on a Chess board, and with simple rules, but normally a player with only a few of games experience can beat the best computer AIs.

At a conceptual level, tree search algorithms generally rely on the searching exhaustively to a certain depth. While with various optimizations the search will not, in reality, be an exhaustive search, the approach is unlikely to be mimicking a human approach. Furthermore, at leaf nodes of such a search the state is usually evaluated with heuristics hand-crafted by the AI designer for the specific game or problem.

These approaches do not indicate how higher-level concepts might be extracted from simple rules of the game, or how structured strategies might be identified by a human. For example, given a Chess position a human might consider two strategies at a given moment (e.g. 'attack opponent queen' or 'defend my king') before considering which moves in particular to use to enact the chosen strategy. Tree search approaches do not operate on a level which either presupposes or provides conceptual game structures (the human-made AI heuristics may, of course, incorporate them, but this is then an explicit proviso by the human AI designer).

More recently, Monte Carlo Tree Search (MCTS) algorithms (7) have been developed which overcome a number of the limitations of the more traditional tree search approaches.

T.Anthony, D. Polani and C.L.Nehaniv are with the Adaptive Systems Research Group, School of Computer Science, University of Hertfordshire, Hatfield, Herts, AL10 9AB, U.K. e-mail: (research@tomanthony.co.uk,{D.Polani,C.L.Nehaniv}@herts.ac.uk).

[1] recent progress in Go-playing AI may render McCarthy's pessimistic prediction concerning performance moot, but, the qualitative criticism stands.

MCTS algorithms represent an important breakthrough in themselves, and lead us to a better understanding of tree searching. However, whilst MCTS has significantly extended the potential ability of tree search algorithms, it remains limited by similar conceptual constraints as previous tree search methods.

In the present paper we propose a model that we suggest is more cognitively plausible and yet also provides first steps towards novel methods which could help address the weaknesses of tree search (and may be used in alongside them). The methods we present have arisen from a different line of thought than MCTS and tree search in general. As this is - to our knowledge - the first application of this train of thought to games, at this early stage it is not intended to out-compete state-of-the-art approaches in terms of performance, but rather to develop qualitatively different, alternative approaches which with additional research may help to improve our understanding and approach to game playing AIs.

The model we propose stems from cognitive and biological considerations, and for this purpose we adopt the perspective of intelligence arising from situatedness and embodiment (8) and view the AI player as an agent that is 'embodied' within an environment (9; 10). The agent's actuator options will correspond to the legal moves within the game, and its sensors reflect the state of the game (those parts available to that player according to the relevant rules).

Furthermore, we create an incentive towards structured decisions by imposing a cost on the search/decision process; this is closely related to the concept of *bounded rationality* (11; 12) which deals with decision making when working with limited information, cognitive capacity, and time and is used as a model of human decision-making in economics (13). As natural cost functionals for decision processes, we use information-theoretical quantities; there is a significant body of evidence that such quantities have not only a prominent role in learning theory (14; 15), but also that various aspects of biological cognition can be successfully described and understood by assuming informational processing costs being imposed on organisms (16; 17; 18; 19; 20; 21; 22).

Thus, our adoption of an information-theoretic framework in the context of decisions in games is plausible not only from a learning- and decision-theoretic point of view, but also from the perspective of a biologically oriented high-level view of cognition where pay-offs conferred by a decision must be traded off with the informational effort of achieving them.

Here, more specifically, we combine this "thinking in informational constraints" with *empowerment* (23; 24), another information-theoretic concept generalising the notion of 'mobility' (25) or 'options' available to an agent in its environment. Empowerment can be intuitively thought of as a measure of how many observable changes an embodied agent, starting from its current state, can make to his environment via its subsequent actions. Essentially, it is a measure of mobility that is generalized as it can directly incorporate randomness as well as incomplete information without any changes to the formalism. If noise causes actions to produce less controllable results, this is detected via a lower empowerment value.

This allows one to treat stochastic systems, systems with incomplete information, dynamical systems, games of complete information and other systems in essentially the same coherent way (26).

In game terms, this above technique could be thought of as a type of 'proto-heuristic' that transcends specific game dynamics and works as a default strategy to be applied, before the game-specific mechanics are refined. This could prove useful either independently or as heuristics from genesis which could be used to guide an AI players behaviour in a new game whilst game-specific heuristics were developed during play. In the present paper we do not go as far as exploring the idea of building game-specific heuristics on top of the proto-heuristics, but focus on deploying the method to generate useful behaviour primitives. We demonstrate the operation of proto-heuristics in two game scenarios and show that intuitively 'sensible' behaviours are selected.

B. Information Theory

To develop the method, we require Shannon's theory of information for which we give a very basic introduction. To begin we introduce *entropy*, which is a measure of uncertainty; the entropy of a variable A is defined as:

$$H(A) = -\sum_{a \in A} p(a) \log p(a). \tag{1}$$

where $p(a)$ is the probability that A is in the state a. The logarithm can be taken to any chosen base; in our paper we always use 2, and the entropy is thus measured in *bits*. If S is another random variable jointly distributed with A, the *conditional entropy* is:

$$H(S|A) = -\sum_{a \in A} p(a) \sum_{s \in S} p(s|a) \log p(s|a). \tag{2}$$

This measures the remaining uncertainty about the value of S, if we know the value of A. This also allows us to measure the *mutual information* between two random variables:

$$\begin{aligned} I(A;S) &= H(S) - H(S|A) \\ &= \sum_{a \in A} \sum_{s \in S} p(a,s) \log \left(\frac{p(a,s)}{p(a)\, p(s)} \right) \end{aligned} \tag{3}$$

Mutual information can be thought of as the reduction in uncertainty about one random variable, given that we know the value of the other. In this paper we will also examine the mutual information between a particular value of a random variable with another random variable:

$$I(a;S) = p(a) \sum_{s \in S} p(s|a) \log \left(\frac{p(a,s)}{p(a)\, p(s)} \right). \tag{4}$$

This can be thought of as the 'contribution' by a specific action a to the total mutual information, and will be useful for selecting a subset of A that maximises mutual information.

Finally, we introduce the information-theoretic concept of the *channel capacity* (27). It is defined as:

$$C(p(s|a)) = \max_{p(a)} I(A;S). \tag{5}$$

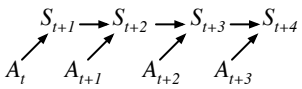


Fig. 1: Bayesian network representation of the perception-action loop.

Channel capacity is measured as the maximum mutual information taken over all possible input (action) distributions, $p(a)$, and depends only on $p(s|a)$, which is fixed for a given starting state s_t. This corresponds to potential maximum amount of information about its prior actions an agent can later observe. One algorithm that can be used to find this maximum is the iterative Blahut-Arimoto algorithm (28).

C. Empowerment

Empowerment, based on the information-theoretic perception-action loop formalism introduced in (24, 29), is a quantity characterizing the "sensorimotor adaptedness" of an agent in its environment. It quantifies the ability of situated agents to influence their environments via their actions.

For the purposes of puzzle-solving and game-play, we translate the setting as follows: consider the player carrying out a move as a sender sending a message, and observing the subsequent state on the board as receiving the response to this message. In terms of Shannon information, when the agent performs an action, it 'injects' information into the environment, and subsequently the agent re-acquires part of this information from the environment via its sensors. Note that in the present paper, we discuss only puzzles and games with perfect information, but the formalism carries over directly to the case of imperfect information game.

For the scenarios relevant in this paper, we will employ a slightly simplified version of the empowerment formalism. The player (agent) is represented by a Bayesian network, shown in Fig. 1, with the random variable S_t the state of the game (as per the player's sensors), and A_t a random variable denoting the action at time t.

As mentioned above, we consider the communication channel formed by the action, state pair A_t, S_{t+1} and compute the channel capacity, i.e. the maximum possible Shannon information that one action A_t can 'inject' or store into the subsequent state S_{t+1}. We define empowerment as this 'motori-sensor' channel capacity:

$$\mathfrak{E} = C(p(s|a)) = \max_{p(a)} I(A; S). \tag{6}$$

If we consider the game as homogenous in time, we can for simplicity ignore the time index, and empowerment only depends on the actual state s_t.

Instead of a single action, it makes often sense to consider an action sequence of length $n > 1$ and its effect on the state. In this case, which we will use throughout most of the paper, we will speak about n-step empowerment. Formally, we first construct a compound random variable of the next n actuations $(A_t,\ A_{t+1},\ A_{t+2},\ \ldots\ ,\ A_{t+n})$ = A_t^n. We now maximize the mutual information between this variable and the state at time $t+n$, represented by S_{t+n}. n-step empowerment is the channel capacity between these:

$$\mathfrak{E} = C(p(s_{t+n}|a_t^n)) = \max_{p(a_t^n)} I(A_t^n; S_{t+n}). \tag{7}$$

It should be noted that $\mathfrak{E}$ depends on S_t (the current state of the world) but to keep the notation unburdened, we will always assume conditioning on the current state S_t implicitly and not explicitly write it.

In the present paper we will present two extensions to the empowerment formalism which are of particular relevance for puzzles and games. The first, discussed in section IV, is impoverished empowerment; it sets constraints on the number of action sequences an agent can retain, and was originally introduced in (30). The second, presented in section VI, introduces the concept of a soft horizon for empowerment which allows an agent to use a 'hazy' prediction of the future to inform action selection. Combined these present a model of resource limitation on the actions that can be retained in memory by the player and corresponds to formulating a 'bounded rationality' constraint on empowerment fully inside the framework of information theory.

Prior to the present paper, and (30), empowerment as a measure has solely been used as a utility applied to *states* but in the present paper we introduce the notion of how empowered an action is. In this case *empowered* corresponds to how much a particular action of action sequence contributes towards the empowerment of a state.

Note that we use the Bayesian network formalism in its causal interpretation (31), as the action nodes have a well-defined interventional interpretation — the player can select its action freely. The model is a simpler version of a more generic Bayesian network model of the perception-action loop where the state of the game is not directly accessible, and only partially observable via sensors. The empowerment formalism generalizes naturally to this more general case of partial observability, and can be considered both in the case where the starting state is externally considered ("objective" empowerment landscape) or where it can only be internally observed (i.e. via context, i.e. distinguishing states by observing sensor sequences for which empowerment values will differ see 32; 26). In fact, the empowerment formalism could be applied without change to the more generic *Predictive State Representation* formalism (PSR, see 33)[2].

Here, however, to develop the formalism for self-motivation and strategy identification, we do not burden ourselves with issues of context or state reconstruction, and we therefore concentrate on the case where the state is fully observable. Furthermore, we are not concerned here with with learning the dynamics model itself, but assume that the model is given (which is, in the game case, typically true for one-player games, and for two-player games one can either use a given opponent model or use, again, the empowerment principle to propose a model for the opponent).

[2]Note that this relies on the actions in the entries of the system-dynamics matrix as being interpreted interventionally (i.e. as freely choosable by the agent) in PSR.

Previous results using empowerment in Maze, Box Pushing, and Pole Balancing scenarios demonstrate that empowerment is able to differentiate the preferable (in terms of mobility) states from the less preferable ones (24), correlates strongly with the graph-theoretic measure of closeness centrality (34) in compatible scenarios, and successfully identified the pole being perfectly upright as amongst the most empowered states in various balancing scenarios (26; 35).

II. Related Work

The idea that artifical agents could derive "appropriate" behaviour from their interaction with the environment was implicit already in early work in cybernetics. However, concrete initiatives on how to make that notion precise arose as a consequence of renewed interest in neural controllers, for instance, in the first modern model of artificial curiosity (36). The idea that AI could be applied to generic scenarios with the help of intrinsic motivation models has led to a number of approaches in the last decade. The *autotelic principle* aims at identifying mechanisms which balance skill and challenge as a mechanism for an agent to improve intrinsically (37). Concrete realizations of that are incarnated as *learning progress*, where the progress in acquiring a model of the environment is considered as the quantity to maximize (38); Schmidhuber's *compression progress* framework (39) which bases its measure directly on the progress in compression efficiency in a Kolmogorov-type framework as actions and observations of an agent proceed through time, and which has been extended towards Reinforcement Learning-like frameworks (*AIXI*, 40).

In (41), the model of *intrinsic rewards* emerging from saliency detectors is adopted (which, in turn, may arise in biological agents from evolutionary considerations), and the *infotaxis* exploration model (considered in a biologically relevant scenario) uses (Shannon) information gain about a navigation target as driver for its exploratory behaviour (42).

The notion of predictive information is used in (43) to drive the behaviour of autonomous robots; it is an information-theoretic generalization of the *homeokinesis* principle as to maintain a predictable, but rich (i.e. non-trivial) future behaviour. An overview over a number of principles important for intrinsic motivation behaviours can be found in (44).

Most of the above principles for intrinsic motivation are process-oriented, i.e. they depend both on the environment as well as on the trajectory of the agent through the environment. The latter, in turn, depends on the learning model. In the case of compression progress and *AIXI*, the prior assumptions are relatively minimal, namely a Turing-complete computational model, but true independence from the learning model is only achieved in the asymptotic case.

Infotaxis, as an explorational model, only relies on a model of the environment to induce a locally information-optimal behaviour. Similarly, empowerment does not require any assumptions about learning models; it is not process-oriented, but state-oriented: assume a particular world and agent embodiment, and assume a given empowerment horizon; then, a given state in the world has a well-defined empowerment value, independently of how the agent travels through the world. In particular, empowerment is emphatically not an world exploration model. Though there are some exploration algorithms for model building which are suitable to be plugged into the empowerment computation, the model acquisition phase is conceptually disparate from the empowerment principle at present and we are not aware of a combined treatment of both in a single coherent framework.

In this paper, we generally assume the world and world dynamics to be essentially known. Similar to the intrinsic reward principle by (41), there is the core assumption of an evolutionary "background story" for the relevance of empowerment for a biological organism, but, different from it, empowerment does not assume dedicated saliency detectors, but works on top of the regular perception-action cycle.

III. General Game Playing

In summary, the scenarios reviewed above indicate that empowerment is able to provide a default utility which 1. derives only from the structure of the problem itself and not from an external reward 2. identifies the desirability of states in way that matches intuition and 3. carries over between scenarios of apparently different character.

This makes it a promising candidate to assign a proto-utility to states of a given system, even before a utility (and a goal) have been explicitly specified.

Importantly, empowerment is more than a naive mobility measure; in calculating empowerment for a given state, it incorporates the structure and dynamics of the agent's world and embodiment. In an abstract game scenario, it would be in principle possible to attribute arbitrary labels to actions in different states. However, in biology, there is some evidence that available actions of an organism evolved to match the ecological niche of the organism and simplify its interaction with its environment (8; 45). We propose that a similar match of action set and game dynamics may also be typical for games that humans find attractive to play (similar to the issue of predictability of games (46)); this hypothesis is the basis for us transferring the empowerment formalism from biological models to game-playing.

We believe that empowerment can help move towards a method that could be used for game playing in general [3], there are three primary issues we must first address:

1) There are reasons to suspect that the ability of biological cognition to structure its decision-making process is driven by the necessity to economize its information processing (48). In other words, we postulate that suitable bounded rationality assumptions are necessary to generate structured behaviour. We will represent these assumptions entirely in terms of the language of our information-theoretic framework, in terms of limited 'informational bandwidth' of actions. For games this cognitive cost to processing the environment is especially true where we desire an AI player to play in real-time or at least as fast as a human player.

[3] the problem of "game playing in general" might include, but is not limited to the Stanford AAAI General Game Playing competition (47)

2) For n-step empowerment to be effective in most scenarios, including games, the reliance on a strict horizon depth is problematic and needs to be addressed.
3) The action policy generated by empowerment should identify that different states have different utilities. Naive mobility-like empowerment does not account for the fact that being able to reach some states can be more advantageous that being able to reach others.

In sections IV we will address the first issue. As for issue 2 and 3, it turns out that they are very related to one another; they will be discussed further in sections V and VI.

Finally, in section VII we will bring together all considerations and apply it to a selection of game scenarios.

IV. Impoverished empowerment

In the spirit of bounded rationality outlined above, we modified the n-step empowerment algorithm to introduce a constraint on the bandwidth of action sequences that an agent could retain. We call this modified concept 'impoverished empowerment' (30). This allows us to identify possible favourable trade-offs, where a large reduction in the bandwidth of action sequences has little impact of empowerment.

While in the original empowerment definition, all possible action sequences leading to various states are considered, in impoverished empowerment, one considers only a strongly restricted set of action sequences. Therefore, we need to identify action sequences which are most empowering, i.e those contributing most to the agent's empowerment; how one action sequence can be more empowering than another is a function of the action sequence's stochasticity (does it usually get where it wanted to go), and whether other action sequences lead to the same state (are there other ways to get there).

A. Scenario

To investigate the impoverished empowerment concept we revisited the scenario from (24); a player is situated within a 2-dimensional infinite gridworld and can select one of 4 actions (North, South, East, and West) in any single time step. Each action moves the agent by one space into the corresponding cell, provided it is not occupied by a wall. The state of the world is completely determined by the position of the agent.

B. Impoverished Empowerment Algorithm

This bandwidth reduction works by clustering the available action sequences together into a number of groups, from which a single representative action sequence is then selected. The selected action sequences then form a reduced set of action sequences, for which we can calculate the empowerment.

Stage 1

Compute the empowerment in the conventional way, obtaining a empowerment-maximizing probability distribution $p(a_t^n)$ for all n-step action sequences a (typically with $n < 6$).

Having calculated the empowerment we have two distributions: $p(a_t^n)$ is the capacity achieving distribution of action sequences and $p(s_{t+n}|a_t^n)$ is the channel that represents the results of an agent's interactions with the environment. For conciseness we will write A to represent action *sequences*.

Stage 2

In traditional empowerment computation, $p(a_t^n)$ is retained for all n-step sequences a. Here, however, we assume a bandwidth limitation on how many such action sequences can be retained. Instead of 'remembering' $p(a_t^n)$ for all action sequences a, we *impoverish* $p(a_t^n)$. i.e. we are going to 'thin down' the action sequences to the desired bandwidth limit.

To stay entirely in the information-theoretic framework, we employ the so-called *information bottleneck* method (49; 50). Here, one assumes that the probability $p(s_{t+n}|a_t^n)$ is given, meaning you need a model of what will be possible outcomes for a given action by a player in a given state. In single player games this is easily determined, whereas in multiplayer games we need a model of the other players (we discuss this more in section IX-A).

We start by setting our designed bandwidth limit by selecting a cardinality for a variable G where $|G| \leq |A_t^n|$; we now wish to find a distribution $p(g|a_t^n)$, where g is a group of action sequences with $g \in G$.

The information bottleneck algorithm (see appendix B) can be used to produce this mapping, using the original channel as an input. It acts to minimise $I(G; A_t^n)$ while keeping $I(S_{t+n}; G)$ constant; it can be thought of 'squeezing' the information A_t^n shares with S_{t+n} though the new variable G to maximize the information A_t^n shares with S_{t+n} whilst discarding the irrelevant aspects. By setting a cardinality for G and then running the information bottleneck algorithm we obtain a conditional distribution $p(g|a_t^n)$, which acts as a mapping of actions to groups.

The result of this is action sequences that usually lead to identical states are clustered together to into groups. However, if the number of groups is less than the number of observed states then beyond the identical state action sequences, the grouping is arbitrary, as seen in Fig. 2. This is because there is nothing to imply any relation between states, be it spatial or otherwise - states are only consistently grouped with others that lead to the same state.

Contrary to what might be expected, introducing noise into the environment actually improves the clustering of actions to those that are more 'similar' (in this case spatially). This is due to the possibility to be 'blown off course', meaning the agent sometimes ends up not in the expected outcome state but in a nearby one which results in a slight overlap of outcome states between similar action sequences. However, it is clear that relying on noise for such a result is not ideal and a better solution to this problem is introduced in section VI.

Stage 3

Because our aim is to select a subset of our original action sequences to form the new action policy for the agent, we must use an algorithm to 'decompose' this conditional distribution $p(g|a_t^n)$ into a new distribution of action sequences, which has an entropy within the specified bandwidth limit.

We wish to maximize empowerment, so for each g we select the action sequence which provides the most towards our empowerment (i.e. the highest value of $I(a_t^n; S_{t+n}|g)$). However, when selecting a representative action sequence for a given g we must consider $p(g|a_t^n)$ (i.e. does this action sequence truly represent this group) so we weight on that;

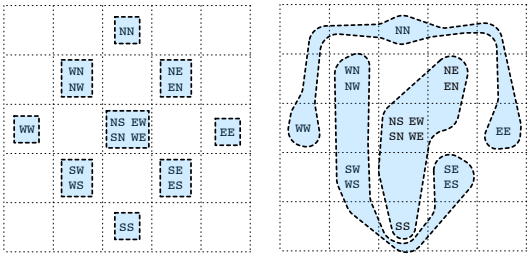


Fig. 2: Visualization of action sequence grouping using Impoverished Empowerment in an empty gridworld with 2-steps; each two character combination (e.g. NW) indicates a 2-step action sequence that leads to that cell from the center cell. Lighter lines represent the grid cells, darker lines the groupings.

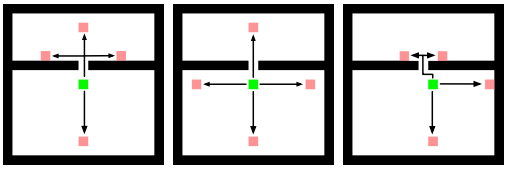


Fig. 3: Typical behaviours where 4 action sequences were selected from 4^6 possibilities. The agent's starting location is shown in green, and its various final locations in pink.

however in most cases the mapping between g and a_t^n is a hard partitioning so this is not normally important. This results in collapsing groups to their 'dominant' action sequence.

C. Impoverishment Results

Figure 3 shows three typical outcomes of this algorithm; in this example we have a bandwidth constraint of 2 bits corresponding to 4 action sequences, operating on sequences with a length of 6 actions; this is a reduction of $4^6 = 4096$ action sequences down to 4. The walls are represented by black, the starting position of the agent is the green center square, and the selected trajectories by the thin arrowed lines with a pink cell marking the end location of the sequence. The result that emerges consistently for different starting states is a set of 'skeleton' action sequences set extending into the state space around the agent. In particular, note that stepping through the doorway which intuitively constitutes an environmental feature of particular salient interest is very often found amongst the 4 action sequences.

Inspection reveals that a characteristic feature of the sequences surviving the impoverishment is the end points of each sequence usually each have a single unique sequence (of the available 4^6) that reaches them.

This can be understood by the following considerations: In order to maintain empowerment whilst reducing bandwidth, the most effective way is to eliminate equivalent actions first since these 'waste' action bandwidth without providing a richer set of end states. States reachable by only one action sequence are therefore advantageous to retain during impoverishment; in Fig. 3, the last action sequences the agent will retain are those leading to states that have only a single unique sequence that reaches them. This is a consequence of selecting the action from each group by $I(a_t^n; S_{t+n})$, and may or may not be desirable; however, in the soft-horizon empowerment model to follow we will see this result disappears.

V. The Horizon

Identifying the correct value for n, for n-step empowerment (i.e. the empowerment horizon depth) is critical for being able to make good use of empowerment in an unknown scenario. A value of n which is too small can mean that states are not correctly differentiated from one another as some options lie beyond the agents horizon. By contrast a value of n which is too large (given the size of the world) can allow the agent to believe all states are equally empowered (30).

Furthermore, it is unlikely that a static value of n would be suitable in many non-trivial scenarios (where different parts of the scenario require different search horizons), and having a 'hard' horizon compounds this.

A. Softening the horizon

We understand intuitively that, when planning ahead in a game, a human player does not employ a hard horizon, but instead probably examines some moves ahead precisely, and beyond that has a somewhat hazy prediction of likely outcomes.

In the soft-horizon empowerment model we use a similar 'softening' of the horizon, and demonstrate how it also helps identify relationships between action sequences which allows the previously presented clustering process to operate more effectively. It allows us to group sets of action sequences together into 'alike' sequences (determined by the overlap in their potential future states), with the resulting groups of action sequences representing 'strategies' . This will be shown later to be useful for making complex puzzles easier to manage for agents. Furthermore, we will show that this horizon softening can help to estimate any ongoing utility we may have in future states, having followed an action sequence. We acknowledge that some future states may be more 'empowered' than others (i.e. lead on to states with more empowerment).

VI. 'Soft-horizon' Empowerment

Soft-horizon empowerment is an extension of the impoverished empowerment model and provides two significant improvements: the clustering of action sequences into groups is enhanced such that the clusters formed represent *strategies*, and it allows an agent to roughly forecast future empowerment following an action sequence. We will show that these features also suggest a solution to having to pre-determine the appropriate horizon value, n, for a given scenario.

A. Split the horizon

Again, we designate a set of actions A, which an agent can select from in any given time step. For convenience we label the number of possible actions in any time step, $|A|$, as c.

We form all possible action sequences of length n, representing all possible action 'trajectories' a player could take in n time steps, such that we have c^n trajectories.

From here, we can imagine a set of c^n possible states the agent arrived in corresponding to the trajectory the agent took, S_{t+n}. It is likely that there are less than c^n unique states, because some trajectories are likely commutative, the game world is Markovian and also because the world is possibly stochastic, but for now we proceed on the assumption that c^n trajectories leads to c^n states (we show how to optimize this in section (30)).

Next, we consider from each of these c^n states what the player could do in an additional m time steps, using the same action set as previously.

We now have for every original trajectory c^n, a set of c^m possible ongoing trajectories. From the combination of these we can create a set of states that represents the outcomes of all trajectories of $n+m$ steps, and label this S_{t+n+m}.

We can form a channel from these states and actions; traditional empowerment's channel would be $p(s_{t+n}|a_t^n)$, corresponding colloquially to 'what is the probability of ending up in a certain state given the player performed a certain action'. With the two trajectories we could form the channel $p(s_{t+n+m}|a_t^{n+m})$, which would be equivalent to if we had simply increased n by m additional steps.

Instead, we create a channel $p(s_{t+n+m}|a_t^n)$, corresponding colloquially to 'what is the probability of ending up in a certain state in $n+m$ steps time if the player performs a given action sequence in the first n steps'. Essentially we are forecasting the potential ongoing future that would follow from starting with a given n-step action sequence.

To do this we need to aggregate and normalise the various distributions of S_{t+n+m} for those which stem from the same original n-step action sequence, a_t^n (their common 'ancestor sequence'). We can calculate this channel:

$$p(s_{t+n+m}|a_t^n) = \frac{\sum_{A_{t+n}^m} p(s_{t+n+m}|a_t^n, a_{t+n}^m)}{\left|A_{t+n}^m\right|} \tag{8}$$

where

$$p(s_{t+n+m}|a_t^{n+m}) \equiv p(s_{t+n+m}|a_t^n, a_{t+n}^m) \tag{9}$$

The result of this 'folding back' to ancestor sequences is that the channel now incorporates two important aspects of the initial n-step sequences:

1) each value for a_t^n now has a rough forecast of its future which can be used to approximate a 'future empowerment' value, i.e. what is a players empowerment likely to be after completing the given n-step action sequence, a_t^n.
2) the distribution of potential future states, $S_{t+n+m}|a_t^n$, for different values of a_t^n can be used to compare the potential overlap in the possible futures that follow from those values of a_t^n. This corresponds to how similar they are in terms of strategy, which we call *strategic affinity*.

Point 1 empowers us to differentiate between possible action sequences in terms of utility; naive empowerment is simply counting states whereas this model acknowledges that some potential states will not be as empowered as others. We show how to calculate this forecast of ongoing empowerment in section VI-C.

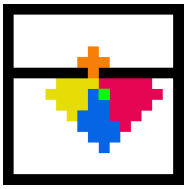


Fig. 4: An example grouping of action sequences, shown here by the colouring of their final states.

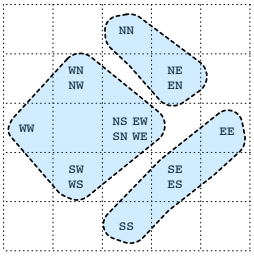


Fig. 5: Visualization of the action sequence grouping from Fig. 2 when the grouping is performed with soft-horizon empowerment. The action sequences now cluster together into 'strategies' formed of similar action sequences that could potentially lead to the same future states.

B. Strategic Affinity

The overlap between the potential futures of each n-step sequence of actions causes them to be grouped together when this channel is fed into the impoverishment algorithm outlined in section IV-B, which brings about the emergence of strategies instead of arbitrary groups previously seen.

The effect of this clustering of action sequences, by their *strategic affinity*, can be illustrated easily in a gridworld as in such an scenario it corresponds closely to geographically close states (see Fig. 5); with more complex worlds such a visualization breaks down but the effect of clustering 'nearby' states remains. An example of such a mapping can be seen in Fig. 4. For many games, this grouping already gives an insight into how tasks may be simplified; either by acting as a coarse representation of the problem or as a tool to identify separate local sub-problems that could be dealt with separately.

Selecting an appropriate bandwidth limit (cardinality) for the number of strategies to be selected is a question not explored in the current paper; we suggest there is rarely a 'correct' answer as selecting a different granularity of strategies will have various different trade-offs.

While soft-horizon empowerment neatly encapsulates strategic affinity, we note that the concept of strategic affinity can be incorporated into other game-playing models, such as MCTS, outside of the empowerment formalism. Combined with a method such as k-means clustering, which does not specify the number of cluster, we hypothesise it would also be possible to use strategic affinity to identify 'natural' strategies.

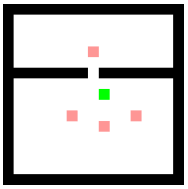


Fig. 6: End states of the 4 action sequences selected to represent the 'strategies' seen in Fig. 4; each is distant from the walls to improve their individual ongoing empowerment.

C. Reducing strategies to actions

We wish to retain only a subset of actions and will use this clustering of action sequences into strategy groups to select a subset of our original action sequences to form the new action policy for the player. To reduce these strategy groups to single action sequences, we select an action sequence from each strategy group that we predict will lead to the most empowered state. We will roughly approximate this forecasted empowerment without actually fully calculating the channel capacity that follows each action sequence.

Earlier we formed the channel $p(s_{t+n+m}|a_t^n, a_{t+n}^m)$ from two separate stages of action sequences, the initial n-steps and the subsequent m-steps. We calculate the channel capacity of this channel which provides a capacity achieving distribution of action sequences $p(a_t^{n+m})$. We now break this channel up into separate channels based on all those where a_t^n is identical, i.e. one channel for each case in which the first n steps are identical.

We now have a set of channels, corresponding to each set of m-step sequences that stem from common ancestor sequences. For each of these 'sub-channels' we sum the mutual information for each sequence a_t^{n+m} in this sub-channel with S_{t+n+m}, using the capacity achieving distribution of action sequences calculated above. More formally, $\sum_{a_{t+n}^m \in A_{t+n}^m} I(a_t^n, a_{t+n}^m; S_{t+n+m})$ where $a_t^{n+m} \equiv a_t^n, a_{t+n}^m$. This gives us an approximation of the $(n+m)$-step empowerment for each sequence of n-steps; we can now select, from within each strategy group, those that are most empowered.

As before we must weight this by their likelihood to map to that g (i.e. the highest value of $p(g|a_t^n)$ for the given g), although once again usually these mappings are deterministic so this is unnecessary.

For the mapping shown in Fig. 4 this leads to the selection of action sequences with the end states shown in Fig. 6.

It can be seen that this results in collapsing strategies to the action sequences which are forecast to lead to the most empowered states. Without any explicit goal or reward, we are able to identify action sequences which represent different strategies, and that are forecast to have future utility. The complete soft-horizon empowerment algorithm is presented appendix A.

D. Single-step iterations to approximate full empowerment

One major problem of the traditional empowerment computation is the necessity to calculate the channel capacity in view of a whole set of n-step actions. Their number grows exponentially with n and this computation thus becomes infeasible for large n.

Fig. 7: A Sokoban inspired gridworld scenario. Green represents the player, and purple represents the pushable box, which is blocking the door.

In (30), we therefore introduced a model whereby we iteratively extend the empowerment horizon by 1 step followed by an impoverishment phase that restricts the number of retained action sequences to be equal to the number of observed states. Colloquially this is summed up as 'only remember one action sequence to reach each state'. This is usually sufficient to ensure the player retains full empowerment whilst significantly improving the computational complexity. With this optimisation the computational bounds on empowerment grow with the number of states (usually linear) instead of with the number of action sequences (usually exponential). Space restrictions prevent us from including the algorithm here, but we have used it for the n-phase of empowerment calculations.

E. Alternative Second Horizon Method

An alternative approach, not explored in the present paper, to using the second horizon to predict the future empowerment is to use an equi-distribution of actions over the m-steps forming the second horizon, as opposed to calculating the channel capacity (step in appendix sec:algoappendix). This approximation is algorithmically cheaper, but at the usually at the expense of a less accurate forecast of future empowerment, as well as a less consistent identification of strategies. However, it may prove to be one path to optimising the performance of soft-horizon empowerment.

VII. Game Scenarios and Results

A. 'Sokoban'

Many puzzle games concern themselves with arranging objects in a small space to clear a path, towards a 'good' configuration. Strategy games often are concerned with route finding and similar such algorithms, and heuristics for these often have to be crafted carefully for a particular game's dynamics.

As an example of such games, we examine a simplified box-pushing scenario inspired by Sokoban. In the original incarnation, each level has a variety of boxes which need to be pushed (never pulled) by the player into some designated configuration; when this was completed, the player completes the level and progresses to the next. Sokoban has received some attention for the planning problems it introduces (51; 52), and most pertinent approaches to it are explicitly search-based and tuned towards the particular problem.

We are changing the original Sokoban problem insofar as that in our scenario there are *no* target positions for the boxes, and in fact there is *no* goal or target at all. As stated, we

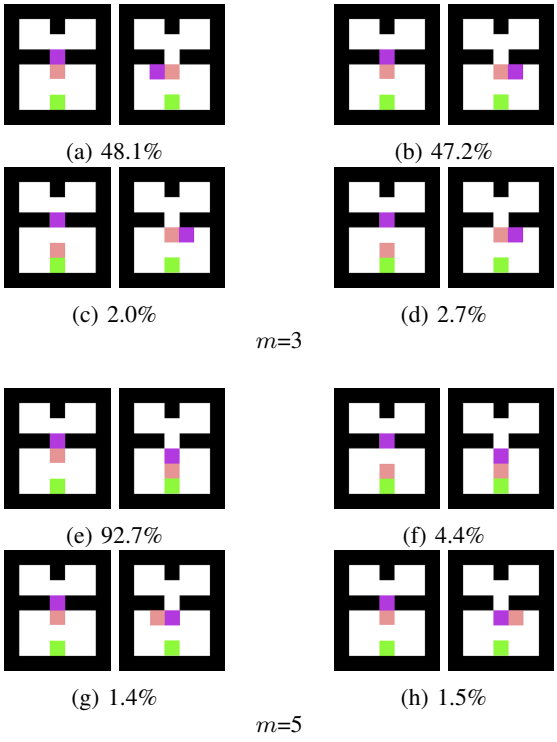


Fig. 8: The distribution of final game states following the selected 4-step action sequences selected by an AI player constrained to 2 action sequences, aggregated from 1000 runs. The pink cells indicate the player's final position.

postulate a critical element for general game playing is self-motivation that precedes the concretization of tasks, thus we adapted the game accordingly.

Figure 7 shows the basic scenario, with a player and a pushable box. Similarly to the earlier gridworlds, the player can move North, South, East and West in any timestep (there is no 'stand still' option). Should the player move into a neighbouring cell occupied by the box, then the box is pushed in the same direction into the next cell; should the destination cell for the box be blocked then neither the player or the box move and the time step passes without any change in the world.

Our intuition would be that most human players, if presented with this scenario and given 4 time steps to perform a sequence of actions on the understanding that an unknown task will follow in the subsequent time steps, would first consider moves that move the box away from blocking the doorway. Humans, we believe, would understand instinctively from observing the setup of the environment, that the box blocks us from the other room and thus moving it gives us more options in terms of places (states) we can reach.

However, it is not obvious how to enable AI players, without explicit goals and no hand-coded knowledge of their environment, to perform such basic tasks as making this identification. Here we approach this task with the fully generic soft-horizon empowerment concept. We use the following parameters: n=4 and m=3, and a bandwidth constraint limiting the player to selecting 2 action sequences from amongst the $4^n = 4^4 = 256$ possible. This constraint was originally selected to see if the AI player would identify both choices for clearing a pathway to the door, and to allow for a possible contrast in strategies.

In Fig. 8 we can see the distribution of final states that were reached. We represent the final states rather than the action sequences that were selected as there are various paths the player can take to reach the same state.

In Fig.8.(a) and Fig.8.(b) we can see the cases that result the majority of the time (95.3%); in one case the box is pushed to the left or right of the door and in the other case the box is pushed into the doorway. The variants in Fig.8.(c) and Fig.8.(d) are almost identical, just the player has moved away from the box one cell.

We can see that two clear strategies emerged; one to clear the path through the door for the player, and a second to push the box through the door (blocking the player from using the doorway). The two options for clearing a path to the doorway (box pushed left or right of the door) are clustered as being part of the same strategy.

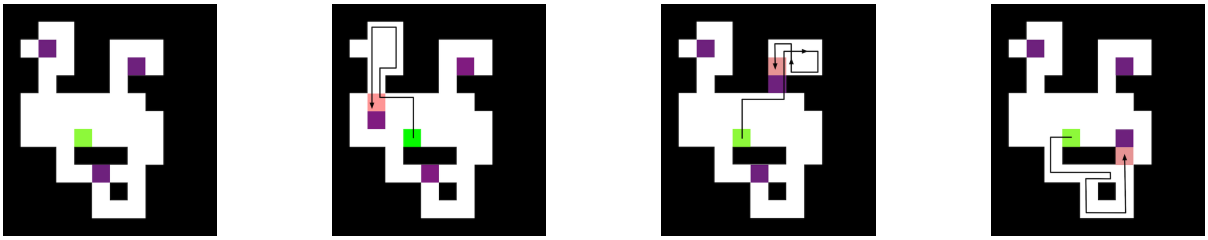
However, it is clear that the types of strategy that can arise, and the ways that action sequences are clustered together, is dependent upon the horizon. If we revisit the same scenario but now adjust the second horizon to be longer, setting $m = 5$ then we can see the results change.

The second row of Fig. 8 show the altered results, and in Fig. 8.(e) we can see that there is a single set of results states that now form the majority of the results (92.7%). We can see that clearing the box to the left or right of the door no longer is part of the strategy; now the player has a horizon of 5 steps it prefers to retain the option of being able to move the box to either the left or the right of the door, rather than committing to one side from the outset. Inspection reveals that occupying the cell below the (untouched) box provides the player with an empowerment of $\mathfrak{E} = log_2 38 = 5.25$ bits rather than $\mathfrak{E} = log_2 31 = 4.95$ bits that would be achieved by clearing the door (in either direction) immediately. The choices that clear the door continue to be clustered together, but the scenario is now 'represented' by the higher empowerment option that emerges with the increased horizon.

Figure 9 shows a more complex scenario, with multiple boxes in the world, all 'trapped' in a simple puzzle. Again, there is no explicit goal; for each of the 3 boxes there exists a single unique trajectory that will recover the box from the puzzle without leaving it permanently trapped. Note that trapping any box immediately reduces the player's ability to control its environment and costs it some degrees of freedom (in the state space) afforded by being able to move the box.

The scenario is designed to present multiple intuitive 'goals', which are attainable only via a very sparse set of action sequences. With a horizon of 14 steps there are 268 million possible action sequences (leading to 229 states), of which 13 full retrieve a box. Note that box 2 (top right) cannot be fully retrieved (pass through the doorway) within 14-steps, and that box 1 (top left) is the only box that could be returned to its starting position by the player.

Table I shows the results when allowing the AI player to select 4 action sequences of 14-steps with an extended horizon



(a) Scenario (b) Box 1 (c) Box 2 (d) Box 3

Fig. 9: A second Sokoban scenario, with multiple boxes in purple and the player's starting position in green. Solutions shown represent the fullest possible recovery for the 3 boxes, along with example action-sequences for recovery. Being 1-step short of these solutions is considered partial recovery (not shown).

	Box 1	Box 2	Box 3
Full	81%	43%	69%
Partial/Full	93%	53%	82%

TABLE I: Percentage of runs in which each of the boxes was recovered. We can see the importance of a long enough horizon; box 2 (which cannot be retrieved completely from the room) is recovered less often than the other boxes.

of m=5 steps, averaged over 100 runs. An explicit count of the different paths to the doorway for each box's puzzle room reveals that there are only 6 action sequences that fully retrieve the box to the main room for each of the top 2 boxes, and only one still for the bottom box.

The results indicate the ability of soft-horizon empowerment to discover actions that lead to improved future empowerment. Furthermore, in every run all 3 boxes were moved in some way, with 35% of cases resulting in all 3 boxes are retrieved, and in 58% at least two boxes are retrieved, leading to indications that the division of the action-sequences into strategies is a helpful mechanism towards intuitive goal identification.

B. Pac-Man-inspired Predator-Prey Maze Scenario

Pac-Man and its variants have been studied previously, included using a tree search approach (53), but the aim of the current paper is not to attempt to achieve the performance of these methods but rather to demonstrate that, notwithstanding their genericness, self-motivation concepts such as soft-horizon empowerment are capable of identifying sensible goals of operation in these scenarios on their own and use these to perform the scenario tasks to a good level.

Thus, the final scenario we present is a simplified predator-prey game based on a simplified Pac-Man model; rather than having pills to collect, and any score, the game is simplified to having a set of ghosts that hunt the player and kill him should they catch him. The 'score', which we will measure, is given simply by the time-steps that the player survives before he is killed; however it is important to note that our algorithm will not be given an explicit goal, rather the implicit aim is simply survival. If humans play the game for a few times, it is plausible to assume (and we would also claim some anecdotal evidence for that) that they will quickly decide that survival is the goal of the game without being told. Choosing survival as your strategy is a perfectly natural decision; assuming no further knowledge/constraints beyond the game dynamics, and a single-player game, anything that may or may not happen later has your survival in the present as its precondition.

In the original Pac-Man game, each ghost uses a unique strategy (to add variation and improve the gameplay) and they were not designed to be ruthlessly efficient; the ghosts in our scenario are far more efficient and all use the same algorithm. Here, in each timestep, the player makes a move (there is no 'do nothing' action, but he can indirectly achieve it by moving towards a neighbouring wall), and then the ghosts, in turn, calculate the shortest path to his new location and move. Should multiple routes have the same distance, then the ghosts randomly decide between them. They penalise a route which has another ghost already on it by adding d extra steps to that route; setting $d = 0$ results in the ghosts dumbly following one another in a chain which is easy for the player. Increasing the value makes the ghosts swarm the player more efficiently. For the present results we use $d = 8$ which is a good compromise between ghost efficiency and giving the player sufficient chance to survive long enough to allow different values for n and m to differentiate.

The maze setup we used is shown in Fig. 10, and the location of the 3 ghosts can be seen. Having only 3 ghosts is another compromise for the same reasons as above; using 4 ghosts usually resulted in the player surviving not long enough to get meaningful variance in the results generated with different parameter sets.

The player has a model of the ghosts' algorithm and thus can predict their paths with some accuracy, and is being allowed 4 samples of their possible future positions (which are stochastic given the possibility that for one or more ghosts the path lengths coincide) for a given move of his. However, once no equal routes are present then 1 sample is perfect information, but once one or more ghosts has one or more equal length paths, then the sampling becomes less accurate and may lose information about the possible future ghost moves.

The game begins with the player's first move, and continues until he is caught by any of the ghosts; at this point the player is 'dead' and is no longer allowed to move. However, there is

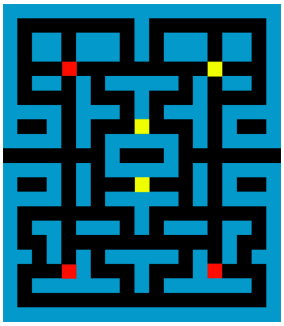


Fig. 10: Pac-Man-inspired scenario, showing the three possible starting positions of the player (yellow) in the center, and of the starting positions of each of the 3 ghosts (red).

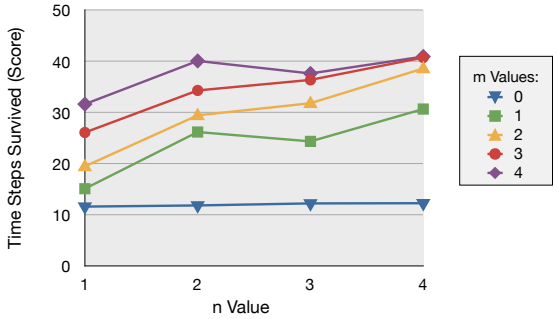


Fig. 11: The player's 'score' for different parameter sets, averaged over 3 different starting positions for the player, and 100 games per data point. It highlights that with no second horizon (m=0) performance does not improve as the first horizon (n) increases.

no special 'death' state in our model; once caught, the player is no longer allowed to move but can still observe the game state (which quickly stops changing anyway, as the ghosts, having swarmed the player, no longer move).

Using the above algorithm, with a cardinality of strategies set to 1 to pick a single action sequence, we observe that the player flees from the ghosts once they come into his horizon; this result from the fact that his future control over the state of the game would drop to zero should he be caught. Death translates directly into the empowerment concept by a vanishing empowerment level. Figure 11 shows the results for various parameter sets, for 3 different starting positions; for each combination of starting position, n value and m value we ran 100 games, then averaged the number of time steps survived over the starting positions for a final average 'score' for each combination of n and m.

Firstly, it can be seen that for $m = 0$, which is equivalent to 'standard' empowerment (23; 24) and does not make use of any features of the algorithm presented that increasing the value of n has no impact on the player's performance. Without a second horizon and thus some measure of his control over the game in the future (beyond the first horizon) there is no pressure to maintain that control. Colloquially, we could say the player only cares about his empowerment in the present moment, not at all about the future. Being able to reach a future state in which he is dead or trapped seems just as good as being able to reach a future state in which he still has a high empowerment; the result is he does not even try to avoid the ghosts and is easily caught.

Once the player has even a small ongoing horizon with $m = 1$ it is easy to see the increase in performance, and with each increase in m performance improves further as the player is better able to predict his future state beyond what action sequence he plans to perform next. For all cases where $m > 0$ it can be seen there is a general trend that increasing n is matched with increasing performance, which would be expected; planning further ahead improves your chances to avoiding the ghosts and finding areas of continued high empowerment.

Note that $n = 2$ performs well and outside of the fit of the other results; this seems, from inspection of individual runs, to be an artefact of the design of the world and the properties of one of the three starting positions, and does not persist that strongly when the starting position is changed. This highlights how a given structure or precondition in a world, which is not immediately observable, could be exploited by specific, hand-crafted AI approaches unique to that exact situation but would be difficult to transfer to other scenarios. The results are shown again, separately for each m value in Fig. 12.

One interesting non-trivial behaviour that consistently emerged from the soft-horizon empowerment algorithm in this scenario was a kiting technique the player would use to 'pull ghosts in'; his employed strategy favoured having a ghost in the immediate cell behind him (this makes that particular ghosts behaviour completely predictable and not only reduces the players's uncertainty about the future but also increases his ability to control it - this includes having a persistent option to commit suicide in a controlled manner at any point). Therefore, the player could often be observed moving back and forth between two cells waiting for a nearby ghost to get to such a position; however, in our observations this did not happen when other ghosts are nearby which would result in the danger of the player being surrounded. This behaviour is not something that would seem intuitive to a human player in this scenario (but humans employ kiting as a method in other games), and whether skirting danger in such a way is desirable in other scenarios is hard to predict.

VIII. Comparison to Mobility

In order to highlight some important differences between soft-horizon empowerment and a greedy mobility algorithm of similar complexity, we present a brief example from the gridworld scenario seen earlier. We created a simple algorithm that samples the world in the same way, and operates with the same goal as soft-horizon empowerment: to select a specified number of actions to maximise utility. The algorithm works thus:

1) Sample the results of performing all possible n-step actions sequences to produce $p(s_{t+n}|a_t^n)$
2) From all reachable states (S_{t+n}), calculate the average mobility (denoted $U(S_{t+n})$) (by sampling) from that state achievable in m-steps

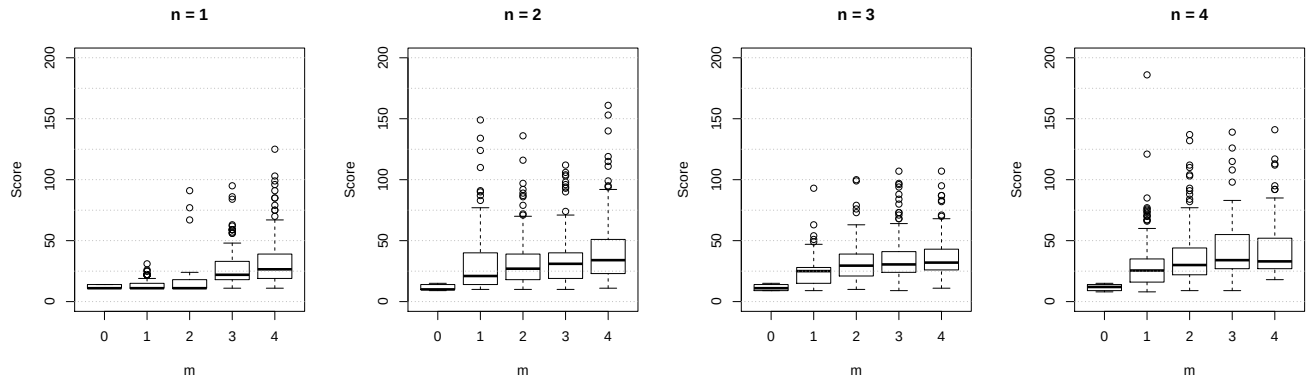


Fig. 12: Boxplot showing quartiles,medians, maximum and minimums for scores across 100 games per pair of horizon values (n,m) for the starting position indicated in Fig 10. The whiskers show 1.5x the interquartile range. It can be observed that a combined horizon of 3-4 is necessary to survive, and then there is a trend towards improved scores as m increases, but as n becomes larger this trend plateaus as performance no longer significantly improves.

Fig. 13: Gridworld scenario. The green cell represents the starting position of the player; the player starts atop a wall.

3) For each value of a_t^n, calculate the average mobility the player would achieve, $p(s_{t+n}|a_t^n) \cdot U(S_{t+n})$
4) Select, according to a specified cardinality, those actions with the highest expected mobility (where there are multiple actions with equally maximal expected mobility, pick a subset randomly)

This results in a set of n-step actions sequences that have an expected mobility over $n+m$-steps in a similar fashion to soft-horizon empowerment, and like empowerment can operate in stochastic scenarios.

A. Gridworld scenario

This gridworld scenario we present here, shown in Fig. 13, operates identically to the gridworld scenarios earlier, but in this instance there is no box for the agent to manipulate. The player starts on a wall along which it cannot move; the player can move east or west and 'fall' into the room on either side, but it cannot then return to the wall. Whilst on the wall the player can effectively stay still by trying to move north or south.

We ran both soft-horizon empowerment and the greedy mobility algorithm with $n = 2$, $m = 2$ and an target action sequence cardinality of 1, such that the output of each algorithm would be a 2-step action sequence selected to maximise empowerment or mobility. For each method, 1000 runs were performed and the results are shown in table II.

All of the actions selected by the greedy mobility algorithm have an expected mobility of 9 moves, and all those moves also lead to a state where $\mathfrak{E} = 3.17$ bits. However, due to the way in which the soft-horizon empowerment algorithm forecasts future empowerment (in the second horizon), it favours moving away from the wall.

Action	Empowerment	Greedy Mobility
EE	51.8%	7.4%
WW	48.2%	7.7%
NN/SS/NS/SN	-	34.2%
EN/ES	-	25.0%
WN/WS	-	25.7%

TABLE II: Distribution of action sequences selected by each method (each over 1000 runs). Action sequences leading to the same state have been grouped. No noise.

Action	Empowerment	Greedy Mobility
EE	-	7.4%
WW	100.0%	8.8%
NN/SS/NS/SN	-	35.5%
EN/ES	-	18.1%
WN/WS	-	30.2%

TABLE III: Distribution of action sequences selected by each method (each over 1000 runs), with noise in the eastern room.

We now introduced some noise into the environment; making it so in the eastern room there was a 50% chance that, for any run, all directions are rotated (N→S, S→E, E→W, W→N). Again, 1000 runs were performed and the results are shown in table III.

The change can be seen clearly; empowerment immediately

adapts and switches to always favouring the western room where it has more control, whereas greedy mobility does not significantly change the distribution of actions it selects. This behaviour is a critical advantage of empowerment over traditional mobility; the rotation of actions does nothing to decrease mobility as each action within a specific run is deterministic. When beginning a new run it is impossible to predict whether the actions would be inverted or not, so whilst there is no decrease in mobility, there is a decrease in *predictability* which negatively impacts empowerment.

Whilst empowerment can be intuitively thought of as a stochastic generalization of mobility, it is actually not exactly the case in many instances; it is possible to encounter stochasticity with no reduction in mobility, but stochasticity is reflected in empowerment due to its reducing of a players control (over their own future).

IX. Discussion

The presented soft-horizon empowerment method exhibits two powerful features, both of which require no hand-coded heuristic:

- the ability to assign sensible 'anticipated utility' values to states where no task or goal has been explicitly specified.
- accounting for the strategic affinity between potential action sequences, as implicitly measured by the overlap in the distribution of their potential future states (naively this can be thought of as how many states that are reachable from A are also reachable from B within the same horizon). This allows a player to select a set of action sequences that fit within a specific bandwidth limit whilst ensuring that they represent a diversity of strategies.

This clustering of action-sequences allows strategic problems to be approached with a coarser grain; by grouping sets of actions together into a common strategy, different strategies can be explored without requiring that every possible action sequence is explored. We believe that such a grouping moves towards a 'cognitively' more plausible perspective which groups strategies a priori according to classes of strategic relevance rather than blindly evaluating an extremely large number of possible moves. Furthermore, by modifying the bandwidth, the concept of having strategies with differing granularities (i.e. 'attack' versus 'attack from the north' and 'attack from the south' etc.) emerges; it has previously been shown that there is a strong urge to compress environments and tasks in such a way (54).

Before, however, we go into a more detailed discussion of the approach in the context of games, some comments are required as to why a heuristic which is based only on the structure of a game and does not take the ultimate game goal into account, can work at all. This is not obvious and seems, on first sight, to contradict the rich body of work on reward-based action selection (grounded in utility theory/reinforcement learning etc.).

To resolve this apparent paradox, one should note that for many games, the structure of the game rules already implicitly encodes partial aspects of the ultimate tasks to a significant degree (similarly to other tasks (55)). For instance, Pac-Man by its very nature is a survival game. Empowerment immediately reflects survival, as a 'dead' player loses all empowerment.

A. Application to Games

In the context of tree search, the ability to cluster action-sequences into strategies introduces the opportunity to imbue game states and corresponding actions with a relatedness which derives from the intrinsic structure of the game and is not externally imposed by human analysis and introspection of the game.

The game tree could now be looked at from a higher level, where the branches represent strategies, and the nodes represent groups of similar states. It is possible to foresee pruning a tree at the level of thus determined strategies rather than individual actions, incurring massive efficiency gains. More importantly, however, these strategies emerge purely from the structure of the game rather than from an externally imposed or assumed semantics.

In many games, it is reasonable to assume having perfect knowledge of transitions in the game state given a move. However, note that the above model is fully robust to the introduction of probabilistic transitions, be it through noise, incomplete information or simultaneous selection of moves by the opponent. The only precondition is is the assumption that one can build a probabilistic model of the dynamics of the system. Such opponent or environment models can be learned adaptively (35; 56). The quality of the model will determine the quality of the generated dynamics, however, we do not investigate this here further.

We illustrated the efficacy of this approach using two scenarios. Importantly, the algorithm was not specifically crafted to suit the particular scenario, but is generic and transfers directly to other examples.

In the Pac-Man-inspired scenario, we demonstrated that acting to maintain anticipated future empowerment is sufficient to provide a strong generic strategy for the player. More precisely, the player, without being set an explicit goal, made the 'natural' decision to flee the ghosts. This behaviour derives from the fact that empowerment is by its very nature a 'survival-type' measure, with death being a 'zero-empowerment' state. With the second horizon's forecast of the future, the player was able to use the essential basic empowerment principle to successfully evade capture for extended periods of time.

We presented several Sokoban-inspired scenarios; the first, smaller, scenario presented a doorway that was blocked by a box, with human intuition identifying clearing the doorway as a sensible idea. We saw that soft-horizon empowerment identified clearing the doorway as a good approach for maximising future utility, and also selected an alternative strategy of pushing the box through the doorway. It was interesting to see that soft-horizon empowerment identified clearing the door with the box to the left or to the right as part of the same strategy, as opposed to pushing the box through the door. This scenario also highlighted how the algorithm differently differentiates strategies based on its horizon limitations.

The second scenario presented 3 trapped boxes each requiring a 14-step action sequence to 'retrieve' from the trap. A

human introspecting the problem could deduce the desirable target states for each box (freeing them so they could be moved into the main room). With a total of 268×10^6 possible action sequences to choose from, and lacking the *a priori* knowledge determining which states should be target states, the algorithm reliably selects a set of action sequences which includes an action sequence for retrieving each of the boxes. Not only are the target states identified as being important but the possible action sequences to recover each of the different boxes are identified as belonging to a different strategy.

The importance of this result lies in the fact that, while again, the approach used is fully generic, it nevertheless gives rise to distinct strategies which would be preferred also based on human inspection. This result is also important for the practical relevance of the approach. The above relevant solutions are found consistently, notwithstanding the quite considerable number and depth of possible action sequences. We suggest that this may shed additional light on how to construct cognitively plausible mechanisms which would allow AI agents to preselect candidates for viable medium-term strategies without requiring full exploration of the space.

The final Sokoban example introduced noise into the environment and compared empowerment to a simple mobility algorithm. It highlighted a distinct advantage of empowerment over mobility in that empowerment identifies a reduction in control and is able to respond appropriately.

B. Relation to Monte-Carlo Tree Search

The presented formalism could be thought of, in certain circumstances as just dividing a transition table into two halves and using forecasts of probabilities of encountering states in the second half to differentiate those states in the first half and assign them some estimated utility. The information-theoretic approach allows this to be quantifiable and easily accessible to analysis. However, we believe the technique presented would work using other methodologies and could be combined with other techniques in the medium-term. One important example of where it could be applied would be in conjunction with a Monte-Carlo Tree Search approach, and we would like to discuss below how the formalism presented in this paper may provide pathways to address some weaknesses with MCTS.

MCTS has been seen to struggle with being a global search in problems with a lot of 'local structure' (57). An example for this is a weakness seen in the Go progam Fuego, which is identified as having territorially weak play (58) because of this problem. Some method which clusters of action sequences into strategies, where the stategic affinity 'distance' between the subsequent states is low, might allow for the tree search to partially operate at the level the strategies instead of single actions and this could help in addressing the problem.

The second aspect of MCTS which has led to some criticism is that it relies on evaluating states to the depth of the terminal nodes they lead to in order to evaluate a state. It is possible that the 'folding back' model of empowerment presented in this paper could be used as a method to evaluate states in an MCTS, which may operate within a defined horizon when no terminal states appear within that horizon. In this way the search could be done without this terminal state requirement, and this might allow a better balance between the depth of the search versus its sparsity. This, of course, would make use of the implicit assumption of structural predictability underlying our formalism.

C. Computational Complexity

We are interested in demonstrating the abilities of the soft-horizon method, but in the present paper we did not aim yet for an optimization of the algorithm. As such, the unrefined soft-horizon algorithm is still very time-consuming.

Previously the time complexity of empowerment was exponential with respect to the horizon, n, until the impoverish-and-iterate approach was introduced in (30) (which is linear with the respect to the number of states encountered).

The present paper introduces soft-horizon empowerment and a second horizon m, and currently the time complexity of the soft-horizon algorithm is exponential with respect to m. We have not yet attempted to apply the iterative impoverishment approach to soft-horizon empowerment, but we expect that this or a similar approach would provide significant improvements.

In continuous scenarios, where early empowerment studies used to be extremely time-consuming, recently developed approximation methods for empowerment allowed to reduce computation time by several orders of magnitude (59).

X. Conclusion

We have proposed soft-horizon empowerment as a candidate for solving implicit 'problems' which are defined by the environment's dynamics without imposing an externally defined reward. We argued that these cases are of a type that are intuitively noticed by human players when first exploring a new game, but which computers struggle to identify.

Importantly, it is seen that this clustering of action sequences into strategies determined by their strategic affinity, combined with aiming to maintain a high level of empowerment (or naive mobility in simpler scenarios) brings about a form of 'self-motivation'. It seems that setting out to maximize the agent's future control over a scenario produces action policies which are intuitively preferred by humans. In addition, the grouping of action sequences into strategies ensures that the 'solutions' produced are diverse in nature, offering a wider selection of options instead of all converging to micro-solutions in the same part of the problem at the expense of other parts. The philosophy of the approach is akin to best preparing the scenario for the agent to maximize its influence so as to react most effectively to an as yet to emerge goal.

In the context of general game-playing, to create an AI that can play new or previously un-encountered games, it is critical to shed its reliance on externally created heuristics (e.g. by humans) and enable it to discover its own. In order to do this, we propose that it will need a level of self-motivation and a general method for assigning preference to states as well as for identifying which actions should be grouped into similar strategies. Soft-horizon empowerment provides a starting point into how we may begin going about this.

REFERENCES

[1] S. Margulies, "Principles of Beauty," *Psychological Reports*, vol. 41, pp. 3–11, 1977.

[2] J. von Neumann, "Zur Theorie der Gesellschaftsspiele," *Mathematische Annalen*, vol. 100, no. 1, pp. 295–320, 1928.

[3] C. E. Shannon, "Programming a computer for playing chess," *Philosophical Magazine*, vol. 41, no. 314, p. 256–275, 1950.

[4] A. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal*, vol. 11, pp. 601–617, 1967.

[5] J. McCarthy, "What is Artificial Intelligence?" 2007. [Online]. Available: http://www-formal.stanford.edu/jmc/whatisai/

[6] O. Syed and A. Syed, "Arimaa - a new game designed to be difficult for computers," *International Computer Games Association Journal*, vol. 26, pp. 138–139, 2003.

[7] G. Chaslot, S. Bakkes, I. Szita, and P. Spronck, "Monte-Carlo Tree Search: A New Framework for Game AI," in *AIIDE*, 2008.

[8] R. Pfeifer and J. C. Bongard, *How the Body Shapes the Way We Think: A New View of Intelligence (Bradford Books)*. The MIT Press, 2006.

[9] F. J. Varela, E. T. Thompson, and E. Rosch, *The Embodied Mind: Cognitive Science and Human Experience*, new edition ed. The MIT Press, Nov. 1992.

[10] T. Quick, K. Dautenhahn, C. L. Nehaniv, and G. Roberts, "On Bots and Bacteria: Ontology Independent Embodiment," in *Proc. of 5th European Conference on Artificial Life (ECAL)*, 1999, pp. 339–343.

[11] H. A. Simon, *Models of man: social and rational; mathematical essays on rational human behavior in a social setting*. New York: Wiley, 1957.

[12] O. E. Williamson, "The Economics of Organization: The Transaction Cost Approach," *The American Journal of Sociology*, vol. 87, no. 3, pp. 548–577, 1981.

[13] C. A. Tisdell, *Bounded rationality and economic evolution : a contribution to decision making, economics, and management*. Edward Elgar, Cheltenham, UK, 1996.

[14] D. A. McAllester, "PAC-Bayesian Model Averaging," in *In Proceedings of the Twelfth Annual Conference on Computational Learning Theory*. ACM Press, 1999, pp. 164–170.

[15] N. Tishby and D. Polani, "Information Theory of Decisions and Actions." in *Perception-Reason-Action Cycle: Models, Algorithms and Systems*, V. Cutsuridis, A. Hussain, and J. Taylor, Eds. Springer, 2010 (in press).

[16] F. Attneave, "Some Informational Aspects of Visual Perception," *Psychological Review*, vol. 61, no. 3, pp. 183–193, 1954.

[17] H. B. Barlow, "Possible Principles Underlying the Transformations of Sensory Messages," in *Sensory Communication: Contributions to the Symposium on Principles of Sensory Communication*, W. A. Rosenblith, Ed. The M.I.T. Press, 1959, pp. 217–234.

[18] ——, "Redundancy Reduction Revisited," *Network: Computation in Neural Systems*, vol. 12, no. 3, pp. 241–253, 2001.

[19] J. J. Atick, "Could Information Theory Provide an Ecological Theory of Sensory Processing," *Network: Computation in Neural Systems*, vol. 3, no. 2, pp. 213–251, May 1992.

[20] M. Prokopenko, V. Gerasimov, and I. Tanev, "Evolving Spatiotemporal Coordination in a Modular Robotic System," in *From Animals to Animats 9: 9th International Conference on the Simulation of Adaptive Behavior (SAB 2006), Rome, Italy, September 25-29 2006*, S. Nolfi, G. Baldassarre, R. Calabretta, J. Hallam, D. Marocco, J.-A. Meyer, and D. Parisi, Eds., vol. 4095. Springer, 2006, pp. 558–569.

[21] W. Bialek, I. Nemenman, and N. Tishby, "Predictability, Complexity, and Learning," *Neural Comp.*, vol. 13, no. 11, pp. 2409–2463, 2001.

[22] N. Ay, N. Bertschinger, R. Der, F. Guettler, and E. Olbrich, "Predictive Information and Explorative Behavior of Autonomous Robots," *European Physical Journal B*, 2008.

[23] A. S. Klyubin, D. Polani, and C. L. Nehaniv, "Empowerment: A Universal Agent-Centric Measure of Control," in *Proceedings of the 2005 IEEE Congress on Evolutionary Computation*, vol. 1. IEEE Press, 2005, pp. 128–135.

[24] ——, "All Else Being Equal Be Empowered," in *Advances in Artificial Life: Proceedings of the 8th European Conference on Artificial Life*, ser. Lecture Notes in Artificial Intelligence, M. S. Capcarrère, A. A. Freitas, P. J. Bentley, C. G. Johnson, and J. Timmis, Eds., vol. 3630. Springer, Sep 2005, pp. 744–753.

[25] E. Slater, "Statistics for the chess computer and the factor of mobility," *Information Theory, IRE Professional Group on*, vol. 1, no. 1, pp. 150 –152, Feb 1953.

[26] A. S. Klyubin, D. Polani, and C. L. Nehaniv, "Keep Your Options Open: An Information-Based Driving Principle for Sensorimotor Systems," *PLoS ONE*, vol. 3, no. 12, 12 2008.

[27] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423, July 1948.

[28] R. Blahut, "Computation of Channel Capacity and Rate Distortion Functions," *IEEE Transactions on Information Theory*, vol. 18, no. 4, pp. 460–473, Jul 1972.

[29] A. S. Klyubin, D. Polani, and C. L. Nehaniv, "Organization of the Information Flow in the Perception-Action Loop of Evolved Agents," in *Proceedings of 2004 NASA/DoD Conference on Evolvable Hardware*, R. S. Zebulum, D. Gwaltney, G. Hornby, D. Keymeulen, J. Lohn, and A. Stoica, Eds. IEEE Computer Society, 2004, pp. 177–180.

[30] T. Anthony, D. Polani, and C. L. Nehaniv, "Impoverished Empowerment: 'Meaningful' Action Sequence Generation through Bandwidth Limitation," in *Proc. European Conference on Artificial Life 2009*. Springer, 2009.

[31] J. Pearl, *Causality: Models, Reasoning and Inference*. Cambridge, UK: Cambridge University Press, 2000.

[32] P. Capdepuy, D. Polani, and C. L. Nehaniv, "Constructing

the Basic Umwelt of Artificial Agents: An Information-Theoretic Approach," 2007, pp. 375–383.

[33] S. Singh, M. R. James, and M. R. Rudary, "Predictive State Representations: A New Theory for Modeling Dynamical Systems," in *Uncertainty in Artificial Intelligence: Proceedings of the Twentieth Conference (UAI)*, 2004, pp. 512–519.

[34] T. Anthony, D. Polani, and C. L. Nehaniv, "On Preferred States of Agents: how Global Structure is reflected in Local Structure," in *Artificial Life XI: Proceedings of the Eleventh International Conference on the Simulation and Synthesis of Living Systems*, S. Bullock, J. Noble, R. Watson, and M. A. Bedau, Eds. MIT Press, Cambridge, MA, 2008, pp. 25–32.

[35] T. Jung, D. Polani, and P. Stone, "Empowerment for continuous agent-environment systems," *Adaptive Behavior - Animals, Animats, Software Agents, Robots, Adaptive Systems*, vol. 19, pp. 16–39, February 2011.

[36] J. Schmidhuber, "A possibility for implementing curiosity and boredom in model-building neural controllers," in *Proc. of the International Conference on Simulation of Adaptive Behavior: From Animals to Animats*, J. A. Meyer and S. W. Wilson, Eds. MIT Press/Bradford Books, 1991, pp. 222–227.

[37] L. Steels, "The Autotelic Principle," in *Embodied Artificial Intelligence: Dagstuhl Castle, Germany, July 7-11, 2003*, ser. Lecture Notes in AI, F. Iida, R. Pfeifer, L. Steels, and Y. Kuniyoshi, Eds. Berlin: Springer Verlag, 2004, vol. 3139, pp. 231–242.

[38] P. Oudeyer, F. Kaplan, and V. V. Hafner, "Intrinsic motivation systems for autonomous mental development," *IEEE Transactions on Evolutionary Computation*, vol. 11, pp. 265–286, 2007.

[39] J. Schmidhuber, "Formal Theory of Creativity, Fun, and Intrinsic Motivation (1990-2010)," *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 3, pp. 230–247, 2010.

[40] J. Veness, K. S. Ng, M. Hutter, W. Uther, and D. Silver, "A Monte-Carlo AIXI Approximation," *Journal of Artificial Intelligence Research*, vol. 40, pp. 95–142, 2011.

[41] S. Singh, A. G. Barto, and N. Chentanez, "Intrinsically Motivated Reinforcement Learning," in *Proceedings of the 18th Annual Conference on Neural Information Processing Systems (NIPS)*, Vancouver, B.C., Canada, Dec 2005.

[42] M. Vergassola, E. Villermaux, and B. I. Shraiman, "'Infotaxis' as a strategy for searching without gradients," *Nature*, vol. 445, no. 7126, pp. 406–409, 2007.

[43] R. D. F. G. E. O. Nihat Ay, Nils Bertschinger, "Predictive information and explorative behavior of autonomous robots," *European Journal of Physics: Complex Systems*, 2008.

[44] P.-Y. Oudeyer and F. Kaplan, "How can we define intrinsic motivation?" in *Proceedings of the 8th International Conference on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems, Lund University Cognitive Studies*, M. Schlesinger, L. Berthouze, and C. Balkenius, Eds., 2008.

[45] T. Berger, "Living information theory," *IEEE Information Theory Society Newsletter*, vol. 53, no. 1, p. 1, 2003.

[46] J. Pearl, *Heuristics: intelligent search strategies for computer problem solving*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1984.

[47] M. Genesereth and N. Love, "General game playing: Overview of the AAAI competition," *AI Magazine*, vol. 26, pp. 62–72, 2005.

[48] S. B. Laughlin, R. R. De Ruyter Van Steveninck, and J. C. Anderson, "The metabolic cost of neural information." *Nature Neuroscience*, vol. 1, no. 1, pp. 36–41, 1998.

[49] N. Tishby, F. Pereira, and W. Bialek, "The information bottleneck method," in *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, 1999, pp. 368–377.

[50] N. Slonim, "The Information Bottleneck: Theory And Applications," Ph.D. dissertation, The Hebrew University, 2003.

[51] A. Junghanns and J. Schaeffer, "Sokoban: A Case-Study in the Application of Domain Knowledge in General Search Enhancements to Increase Efficiency in Single-Agent Search," *Artificial Intelligence, special issue on search*, 2000.

[52] D. Dor and U. Zwick, "SOKOBAN and other motion planning problems," *Comput. Geom. Theory Appl.*, vol. 13, no. 4, pp. 215–228, 1999.

[53] D. Robles and S. M. Lucas, "A simple tree search method for playing Ms. Pac-Man," in *CIG'09: Proceedings of the 5th International Conference on Computational Intelligence and Games*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 249–255.

[54] J. Schmidhuber, "Driven by Compression Progress: A Simple Principle Explains Essential Aspects of Subjective Beauty, Novelty, Surprise, Interestingness, Attention, Curiosity, Creativity, Art, Science, Music, Jokes," *CoRR*, vol. abs/0812.4360, 2008.

[55] J. Lehman and K. Stanley, "Abandoning objectives: Evolution through the search for novelty alone," *Evolutionary computation*, vol. 19, no. 2, pp. 189–223, 2011.

[56] P. Capdepuy, D. Polani, and C. L. Nehaniv, "Constructing the Basic Umwelt of Artificial Agents: An Information-Theoretic Approach," in *Proceedings of the Ninth European Conference on Artificial Life*, ser. LNCS/LNAI, F. Almeida e Costa, L. M. Rocha, E. Costa, I. Harvey, and A. Coutinho, Eds., vol. 4648. Springer, 2007, pp. 375–383.

[57] M. Müller, "Challenges in Monte-Carlo Tree Search," 2010, unpublished. [Online]. Available: http://www.aigamesnetwork.org/_media/main:events:london2010-mcts-challenges.pdf

[58] ——, "Fuego-GB Prototype at the Human machine competition in Barcelona 2010: A Tournament Report and Analysis," University of Alberta, Tech. Rep. TR10-08, 2010.

[59] C. Salge, C. Glackin, and D. Polani, "Approximation of Empowerment in the Continuous Domain," *Advances in Complex Systems*, vol. 16, no. 01n02.

APPENDIX A
SOFT-HORIZON EMPOWERMENT COMPLETE ALGORITHM

The soft-horizon empowerment algorithm consists of two main phases. This appendix presents the complete algorithm. In order to make it somewhat independent and concise, it introduces some notation not used in the main paper.

Phase 1 is not strictly necessary, but acts as a powerful optimization by vastly reducing the number of action sequences that need to be analysed. The main contributions presented in this paper are within phase 2.

A. Setup

1) Define a set $\mathcal{A}$ as the list of all possible single-step actions available to the player.
2) Set $n = 0$. Begin with an empty set containing a list of action sequences, $\mathcal{A}^n$.

B. Phase 1

Phase 1 serves to create a set of action sequences $\mathcal{A}^n$ that, most likely, will reach all possible states within n-steps, but will have very few (0 in a deterministic scenario) redundant sequences. In stochastic scenarios that have heterogeneous noise in the environment it may be that those areas are avoided in preference to staying within more stable states, and in these cases you will find there may be some redundancy in terms of multiple action sequences to the same state.

In a deterministic scenario phase 1 can be entirely skipped; the same optimization can be achieved by selecting at random a single action sequence for each state reachable within n-steps (i.e. for each state s select any single action sequence where $p(s|a_t^n) = 1$).

3) Produce an extended list of action sequences by forming each possible extension for every action sequence in $\mathcal{A}^n$ using every action in $\mathcal{A}$; the number of resultant action sequences should equal $|\mathcal{A}^n| \cdot |\mathcal{A}|$. Replace $\mathcal{A}^n$ with this new list and increment n by 1.
    i. Using the channel/transition table, $p(s_{t+n}|a_t^n)$, note the number of unique states (labelled σ) reachable using the action sequences in $\mathcal{A}^n$, always starting from the current state.
4) Produce $p(a_t^n)$ from $\mathcal{A}^n$, assuming an equi-distribution on $\mathcal{A}^n$. Using this, combined with $p(s_{t+n}|a_t^n)$, as inputs to the Information Bottleneck algorithm (we recommend the implementation at (50), pp. 30, see Appendix B). For G, our groups of actions (labelled T in (50)), we set $|G|$ to be equal to σ, such that the number of groups matches the number of observed states. This will produce a mapping, $p(g|a_t^n)$, which will typically be a hard mapping in game scenarios. Select a random value of a from each G (choosing $argmax_{a_t^n} p(g|a_t^n)$ in cases where it is not a hard mapping). Form a set from these selected values, and use this set to replace $\mathcal{A}^n$.
5) Loop over steps 3 and 4 until n reaches the desired length.

C. Phase 2

Phase 2 extends these base n-step sequences to extended sequences of $n + m$-steps, before collapsing them again such that we retain only a set of n-step sequences which can forecast their own futures in the following m-steps available to them.

6) Produce a list of action sequences, $\mathcal{M}$, by forming every possible m-step sequence of actions from the actions in $\mathcal{A}$.
7) Produce an extended list of action sequences, $\mathcal{A}^{n+m}$, by forming each possible extension for every action sequence in the final value of $\mathcal{A}^n$ using every action sequence in $\mathcal{M}$.
8) Create a channel $p(s_{t+n+m}|a_t^{n+m})$ (where $a_t^{n+m} \in \mathcal{A}^{n+m}$) by sampling from the environmental dynamics for our current state (using the game's transition table). For environments with other players, one can use any approximation of their behaviour available and sample over multiple runs or, lacking that, model them with greedy empowerment maximisation based on a small horizon.
9) Calculate the channel capacity for this channel using the Blahut-Arimoto algorithm, which provides the capacity achieving distribution of action sequences, $p(a_t^{n+m})$.
10) Now collapse $p(s_{t+n+m}|a_t^{n+m})$ to $p(s_{t+n+m}|a_t^n)$ by marginalizing over the equally distributed extension of the action sequences:

$$p(s_{t+n+m}|a_t^n) = \frac{\sum_{a_{t+n}^m} p(s_{t+n+m}|a_t^n, a_{t+n}^m)}{\left|A_{t+n}^m\right|}$$

where

$$p(s_{t+n+m}|a_t^{n+m}) \equiv p(s_{t+n+m}|a_t^n, a_{t+n}^m)$$

11) Apply the Information Bottleneck (as in (50), pp. 30, see Appendix B) to reduce this to a mapping of action sequences to strategies, $p(g|a_t^n)$ where G are our groups of action sequences grouped into strategies. Cardinality of G sets how many strategy groups you wish to select.
12) We now need to select a representative action, $a^{(\text{rep})}$, from each group g that maximises approximated future empowerment (and weight this on how well the action represents the strategy, $p(g|a_t^n)$, which is relevant if $p(g|a_t^n)$ is not deterministic):

$$a^{(\text{rep})}\left(p(s_{t+n+m}|a_t^n, a_{t+n}^m), g\right) =$$

$$\arg\max_{a_t^n} \left(p(g|a_t^n) \cdot \sum_{a_{t+n}^m \in A_{t+n}^m} I(a_t^n, a_{t+n}^m; S_{t+n+m}) \right)$$

Where $a_t^{n+m} \equiv a_t^n, a_{t+n}^m$, and using the capacity achieving distribution of action sequences, $p(a_t^{n+m})$, calculated in step 9 above. Note that the mutual information there requires the full channel, but sums over those parts of the channel with the identical n-steps, so algorithmically it is advisable to calculate and store these mutual information values as whilst doing the channel collapse above.

We can now form a distribution of n-step action sequences from the set of values of $a^{(\text{rep})}$ from each action group; these represent a variety of strategies whilst aiming to maximise future empowerment within those strategies.

APPENDIX B
INFORMATION-BOTTLENECK ALGORITHM

The Information Bottleneck algorithm is a variation of rate-distortion, in which the compressed representation is guided not by a standard rate-distortion function but rather through the concept of relevancy through another variable. We wish to form a compressed representation of the discrete random variable A, denoted by G, but we acknowledge that different choices of distortion would result in different representations but it is likely that we have some understanding of what aspects of A we would like to retain and which could be discarded. The Information Bottleneck method seeks to address this by introducing a third variable, S, which is used to specify the relevant aspects of A.

More formally, we wish to compress $I(G;A)$ while maintaining $I(G;S)$. We are looking to retain the aspects of A which are relevant to S, whilst discarding what else we can. We introduce a Lagrange multiplier, β, which controls the trade-off between these two aspects, meaning we seek a mapping $p(g|a)$ which minimises:

$$\mathcal{L} = I(G;A) - \beta I(G;S)$$

The iterative algorithm we present here is from (50).

Input:

1) Joint distribution $p(s,a)$
2) Trade-off parameter β
3) Cardinality parameter σ and a convergence parameter ϵ

Output:

A mapping $p(t|a)$, where $|G| = \sigma$. For the scenarios in this paper this typically a hard mapping, but it is not necessarily so.

Setup:

Randomly initialise $p(g|a)$, then calculate $p(g)$ and $p(s|g)$ using the corresponding equations below.

Loop:

$$1)\, P^{(m+1)}(g|a) \leftarrow \frac{P^{(m)}(g)}{Z^{(m+1)}(a,\beta)} e^{-\beta D_{KL}[p(s|a)\|p(s|g)]}, \forall\, g \in \mathcal{G}, \forall\, a \in \mathcal{A}.$$

$$2)\, P^{(m+1)}(g) \leftarrow \sum_a p(a) P^{(m+1)}(g|a), \forall\, g \in \mathcal{G}.$$

$$3)\, P^{(m+1)}(s|g) = \frac{1}{P^{(m+1)}(g)} \sum_a P^{(m+1)}(g|a) p(a,s), \forall\, g \in \mathcal{G}, \forall\, s \in \mathcal{S}.$$

Until:

$$JS(P^{(m+1)}(g|a), P^{(m)}(g|a)) \leq \epsilon$$

Where JS is the Jensen-Shannon divergence, based on D, the Kullback–Leibler divergence:

$$JS(P \parallel Q) = \frac{1}{2} D(P \parallel M) + \frac{1}{2} D(Q \parallel M)$$

A. Parameter Values

In the present paper, for all scenarios β = 3, and ϵ = 0.0000001.

The algorithm is based on a random initialisation of $p(t|a)$, so it is usually beneficial to use multiple runs of the algorithm and select that with the best result. Throughout the present paper we used 200 runs of the Information Bottleneck algorithm in each instance it was used.